ARTICLE IN PRESS

The Korean Institute of Communications and Information Sciences

Available online at www.sciencedirect.com

ScienceDirect

ICT Express xxx (xxxx) xxx

www.elsevier.com/locate/icte

Multi-agent reinforcement learning based optimal energy sensing threshold control in distributed cognitive radio networks with directional antenna

Thi Thu Hien Pham[a], Wonjong Noh[b],[*], Sungrae Cho[a],[*]

[a] *School of Computer Science and Engineering, Chung-Ang University, Seoul 06974, South Korea*
[b] *School of Software, Hallym University, Chuncheon 24252, South Korea*

Received 27 June 2023; received in revised form 2 December 2023; accepted 5 January 2024
Available online xxxx

Abstract

In CRNs, it is crucial to develop an efficient and reliable spectrum detector that consistently provides accurate information about the channel state. In this work, we investigate a CSS in a fully-distributed environment where all secondary users (SUs) are equipped with directional antennas and make decisions based solely on their local knowledge without information sharing between SUs. First, we establish a stochastic sequential optimization problem, which is an NP-hard, that maximizes the SU's detection accuracy by the dynamic and optimal control of the energy sensing/detection threshold. It can enable SUs to select an available channel and sector without causing interference to the primary network. To address it in a distributed environment, the problem is transformed into a decentralized partially observed Markov decision process (Dec-POMDP) problem. Second, in order to determine the best control for the Dec-POMDP in a practical environment without any prior knowledge of state–action transition probabilities, we develop a multi-agent deep deterministic policy gradient (MADDPG)-based algorithm, which is referred to as MA-DCSS. This algorithm adopts the centralized training and decentralized execution (CTDE) architecture. Third, we analyzed its computational complexity and showed the proposed approach's scalability by the polynomial computational complexity, in terms of the number of channels, sectors, and SUs. Lastly, the simulation confirms that the proposed scheme provides enhanced performance in terms of convergence speed, accurate detection, and false alarm probabilities when it is compared to baseline algorithms.

Keywords: Cognitive radio networks (CRNs); Cooperative spectrum sensing (CSS); Directional antennas; Multi-agent deep deterministic policy gradient (MADDPG); Reinforcement learning (RL)

1. Introduction

The increasing number of network devices has led to a higher demand for additional radio frequency spectrum bands. To tackle this scarcity of wireless resources, CRNs have emerged as a promising solution. These networks enable SUs to opportunistically access licensed spectrum bands from primary users (PUs). However, this approach requires devices to accurately detect and utilize unoccupied spectrum bands while preventing interference, which is challenging due to the dynamic and uncertain wireless environment, including factors like multi-path fading, shadowing, and receiver uncertainty [1]. Hence, cooperative spectrum sensing (CSS) was introduced to solve these problems. There are two primary CSS schemes: centralized and distributed. In the centralized approach, all SUs must follow instructions from a coordination node called the fusion center (FC). The FC collects sensing results from all SUs and combines them to estimate the final state of the sensed channel band. Various techniques have been proposed for this approach. R. Sarikhani et al. [2] introduce a deep reinforcement learning (DRL) based fusing scheme for the FC, equipping it with a convolutional neural network to identify uncorrelated neighbors for cooperation. W. Na et al. [3] explore a centralized CSS environment where they introduce a single FC that can serve multiple SU groups using a control time slot. One of their objectives is to maximize the accuracy of the energy detector for all SUs to make appropriate transmission decisions consistently. While these centralized approaches [2,3] have shown promising results, they suffer from operational costs and the potential bottleneck problem associated with the FC.

To address these issues on the reliance on the FC within the network and operational costs, the distributed CSS approaches

[*] Corresponding authors.
E-mail addresses: ptthien@uclab.re.kr (T.T.H. Pham), wonjong.noh@hallym.ac.kr (W. Noh), srcho@cau.ac.kr (S. Cho).
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2024.01.001

have been studied. A. Gharib et al. [4] propose a novel clustering scheme that selects a leader for each cluster and groups cooperative SUs based on their correlation value. The scheme emphasizes the significance of diversity among the SUs, as it enables them to bring in a wider range of information, ultimately contributing to better sensing outcomes. A. Gao et al. [5] considered a dynamic clustering solution, where each SU has the flexibility to select partners for cooperation after each time step based on their historical sensing results. These solutions [4,5] can be categorized as partially distributed, as they still require SUs to communicate in a control time slot to gather sufficient information for clustering, even without an explicitly considered coordination node. On the other hand, in the fully-distributed environment, X. Tan et al. [6] propose a dynamic spectrum access scheme based on multi-agent reinforcement learning (MARL), where all SUs are trained to select a set of channels to sense, maximizing transmission chances and minimizing collisions. This work, however, considers an ideal scenario where the spectrum sensing technique employed by the SUs always provides accurate results. M. K. Giri et al. [7] investigate a fully-distributed environment, where they focus on the objective of minimizing the miss detection probability in order to reduce interference to the PUs. The false alarm probability yet is not considered in their optimization goal.

On the other hand, many studies have also demonstrated the advantages of employing directional antennas instead of omni-directional antennas within CRNs [8,9]. Directional antennas can focus the radio frequency (RF) energy in a specific direction, which allows the cognitive radios to enhance their sensitivity and reduce their interference with neighboring devices. For example, the results in [3] showed that directional antennas can help SUs achieve higher average throughput and lower energy consumption. However, [3] is a directional antenna-based centralized CSS approach, and few works take advantage of directional antennas in distributed environments. Moreover, since spectrum sensing is often subject to changing conditions and uncertainties [1], DRL can be a well-suited solution for this as it allows the agents to adapt and learn from their own experience over time. Inspired by this, we investigate a directional fully-distributed CSS scheme leveraging multi-agent DRL. The main contributions of this work are as follows:

- We address a challenging NP-hard stochastic sequential optimization problem aimed at maximizing the SU's detection accuracy by the dynamic and optimal control of the energy detection threshold in a directional cognitive radio network. To address it in a distributed environment, we transform it into a Dec-POMDP problem.
- We introduce a MADDPG-based fully-distributed CSS approach, namely MA-DCSS. Our solution leverages the CTDE architecture, enabling SUs to learn collaboratively during training and operate independently afterward.
- Through simulations, we validate the proposed scheme, showcasing its superiority over baseline algorithms. The evaluations emphasize improved convergence speed, detection probability, and reduced false alarm probability.

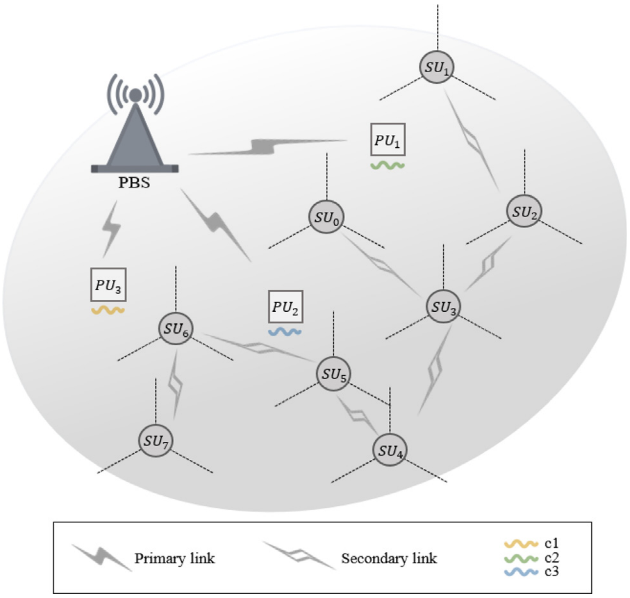

Fig. 1. System model consisting of a primary and a secondary network.

2. System model and problem formulation

2.1. System model

In this study, we take into account a system model that has two networks: a primary network with a primary base station (PBS) and U PUs, as well as a secondary ad-hoc network with M SUs (Fig. 1). Both PUs and SUs are assumed to be static in the networks. In traditional centralized cognitive networks, there is a need for a coordinate node that is in charge of fusing information from other nodes and thereby making decisions. Unlike that, in this system, we focus on a distributed environment where SUs can operate equally after having enough time to collaborate and learn about the environment. Suppose that there are a total of K orthogonal channels that are owned by PUs, i.e., PUs have the first priority to use those channels. For the SUs, since they are unlicensed users, they need to wait for the channels to be set free by the PUs.

All SUs are equipped with directional antennas with L as the number of sectors for each SU and they are ideally non-overlapped. SUs then use their directional antennas to sense the free channels as well as transmit data. Also, with the help of directional antennas, SUs might be able to use the same channel as the PUs without causing interference to the primary network. Meanwhile, PUs are equipped with traditional omni-directional antennas for communication purposes. This network model also was referred to as Omn-Dir-CRN according to [8].

In this work, we assume that the energy detection (ED) based spectrum sensing method is employed at every SU in order to sense the appearance of PUs and decide whether a particular channel is occupied by PUs or not. ED is a non-coherent and very popular detection method since it has no requirement for any historical information [1]. This technique normally goes with binary hypothesis testing, in which

$H_1^i(c^i, s^i)$ and $H_0^i(c^i, s^i)$ relatively denote the presence and absence of PUs under the observation of SU_i when it chooses c^i, s^i as the sensing channel and sector, respectively. Let us say $y^i(n|c^i, s^i)$ is the received signal at the ith SU, then we have:

$$y^i(n|c^i, s^i) = \begin{cases} h^i s(n) + u(n), & H_1^i(c^i, s^i), \\ u(n), & H_0^i(c^i, s^i), \end{cases} \tag{1}$$

where h^i represents the channel gain, $s(n)$ is the signal from PU, and $u(n)$ is the addictive white Gaussian noise (AWGN) with a zero mean. The detection process commences with $y^i(n|c^i, s^i)$ being passed through an ideal band-pass filter in order to limit the noise bandwidth [10]. The output is then squared and integrated over an observation time interval. After integration, the final test statistic for the ith SU can be given as:

$$\lambda^i(c^i, s^i) = \frac{1}{N} \sum_{n=1}^{N} |y^i(n|c^i, s^i)|^2, \tag{2}$$

with N as the number of received samples. This test statistic value is then compared with a detection threshold ϵ^i to decide whether a PU is present or not:

$$\lambda^i(c^i, s^i) \underset{H0}{\overset{H1}{\gtrless}} \epsilon^i(c^i, s^i). \tag{3}$$

The PU occupancy state on a channel is formed as a two-state Markov chain model [11], in which *busy*(1) and *idle*(0) are the two states of a channel. Let us say α and $1-\alpha$ are the probability of the transitions from *busy* to *busy* and *busy* to *idle*, respectively. Then, the probability of the transitions from *idle* to *idle* and *idle* to *busy* are given by β and $1-\beta$ correspondingly. The occupancy state transition probability is the same for all channels, and it can be expressed as the matrix:

$$\begin{aligned} P_{trans} &= \begin{bmatrix} idle \to idle & idle \to busy \\ busy \to idle & busy \to busy \end{bmatrix} \\ &= \begin{bmatrix} \beta & 1-\beta \\ 1-\alpha & \alpha \end{bmatrix}. \end{aligned} \tag{4}$$

With regard to the time frame structure of SUs, since each SU is individually operated, a period of time for controlling purposes is unnecessary in comparison to other centralized systems [2,3]. Each active period of a SU contains two main parts, namely sensing and transmission. Hence, let us say T and τ denoting the length of the active period and the sensing duration, respectively. These parameters remain constant across all SUs in the system.

2.2. Problem formulation

For the ED, regardless of c^i and s^i, the PU detection probability for SU_i can be expressed as below [3]:

$$P_d^i = P\left(\lambda^i > \epsilon^i | H_1\right) = Q_{\frac{N}{2}}\left(\sqrt{2\lambda^i}, \sqrt{\epsilon^i}\right), \tag{5}$$

where $Q_{\frac{N}{2}}(.,.)$ is the generalized Marcum Q function. On the other hand, the false alarm probability for SU_i can be calculated as:

$$P_f^i = P\left(\lambda^i > \epsilon^i | H_0\right) = \frac{\Gamma\left(\frac{N}{2}, \frac{\epsilon^i}{2}\right)}{\Gamma\left(\frac{N}{2}\right)}, \tag{6}$$

where $\Gamma(.)$, and $\Gamma(.,.)$ are the gamma and incomplete gamma function, respectively. The equations in (5) and (6) were widely used with the assumption of the detection threshold ϵ^i being intact over a period of time. However, in this work, ϵ^i is assumed to be a time-varying variable, which means it is stochastic and changes over time. In this case, the detection probability up to the time t_n can be expressed as:

$$P_d^i(t_n) = \frac{\sum_{t=0}^{t_n} \mathbf{1}_{\left\{\lambda_t^i > \epsilon_t^i \middle| c_t^i, s_t^i\right\}}(e_t)}{\sum_{t=0}^{t_n} \mathbf{1}_{\{H_1^i(c_t^i, s_t^i)\}}(e_t)} \tag{7}$$

where $\mathbf{1}_{\{.\}}(e_t)$ in the indicator of an event happening. Also, the false alarm probability up to the time t_n can be calculated as:

$$P_f^i(t_n) = \frac{\sum_{t=0}^{t_n} \mathbf{1}_{\left\{\lambda_t^i > \epsilon_t^i \middle| c_t^i, s_t^i\right\}}(e_t)}{\sum_{t=0}^{t_n} \mathbf{1}_{\{H_0^i(c_t^i, s_t^i)\}}(e_t)} \tag{8}$$

This work aims to determine the best energy detection threshold w.r.t a channel-sector pair at each time step for all SUs. By finding the optimal variables, it aims to maximize the probability of correctly detecting the presence of PU while minimizing the likelihood of false alarms, which are two key performance factors of the sensing scheme. Therefore, the problem at a specific time step t can be formulated as:

$$\begin{aligned} \max_{\boldsymbol{\epsilon}_t = \{\epsilon_t^i(c_t^i, s_t^i)\}} \quad & \sum_{t \geq 0} \sum_{i=1}^{M} P_d^i(t) + (1 - P_f^i(t)) \\ \text{s.t.} \quad & \epsilon_t^i(c_t^i, s_t^i) \geq 0, \\ & \forall c_t^i \in [0, \ldots, K-1], s_t^i \in [0, \ldots, L-1], \end{aligned} \tag{9}$$

where $\epsilon_t^i(c_t^i, s_t^i)$ is the detection threshold at time step t chosen by SU_i for particular channel c_t^i and sector s_t^i. The problem (9) is a stochastic sequential optimization problem, which is an NP-hard problem [12,13]. Therefore, we re-formulate it in the form of MDP and provide its solution in the following section.

3. Proposed solution

3.1. Decentralized partially Markov decision process

Since each of the SUs has no prior knowledge about the environment as well as other SUs' sensing information, any decision given by SU must be only based on its local knowledge. Therefore, the problem can be transformed into a Dec-POMDP [14], which can be defined using a tuple of $\left(\mathcal{M}, \mathcal{S}, \{\mathcal{O}^i\}_{i\in\mathcal{M}}, \{\mathcal{A}^i\}_{i\in\mathcal{M}}, \{r^i\}_{i\in\mathcal{M}}, \mathcal{P}, \{\mathcal{Y}^i\}_{i\in\mathcal{M}}, \gamma\right)$. Where, $\mathcal{M} = \{1, \ldots, M\}$ is the set of all agents; $\mathcal{S}$ is the set representing the actual state of the environment; $\mathcal{O}^i$ is the partial observation space acknowledged by agent i; $\mathcal{A}^i$ is the action space of ith agent, and $\mathcal{A} := \mathcal{A}^1 \times \cdots \times \mathcal{A}^M$ is the set of all agent's actions; $r^i(s, a, s') : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward the agent i receives from the environment when it takes an action a from state s to a new state s'; $\mathcal{P}(s'|s, a) : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$

is the transition probability to the new state, given a known state and action; $\mathcal{Y}^i : \mathcal{S} \rightarrow \mathcal{O}_i$ is the observation channel that maps from the environment's true state to the agent's observation; finally, $\gamma \in [0, 1)$ is the discount factor. The goal of all agents is to find an optimal policy $\mu^i : \mathcal{O}^i \rightarrow \mathcal{A}^i$ that maximizes the expected long-term discounted reward. More detailed explanations on the key components of the Dec-POMDP are provided below.

(1) Actual state of the environment: The state at time step t is defined as $\mathbf{s}_t = \left[\mathbf{B}_t^i\right]_{i\in\mathcal{M}}$, where $\mathbf{B}_t^i = \left[B_t^{i,k}\right]^T_{k\in 0,\dots,K-1}$ is a matrix of size $K \times L$, in which $B_t^{i,k} = \left[b_t^{i,k,l}\right]_{l\in 0,\dots,L-1}$. Here, $b_t^{i,k,l} = 1$ indicates that there is at least one PU using the kth channel and it is located in the area covered by the lth sector of SU_i, and $b_t^{i,k,l} = 0$ otherwise.

(2) Partial observations by SUs: Due to the physical limitation of SU, it is impossible to obtain the fully actual state of the environment. One SU can only sense one channel and sector pair at a time step, it thereby partially observes the environment state. When performing ED, SU is able to estimate the received signal power for the chosen particular pair. The agent's observation can be defined as $\mathbf{o}_t^i = \left[c_t^i, s_t^i, \lambda_t^i\right]$, where c_t^i, s_t^i are the chosen channel and sector indices, and λ_i^t is the received signal power estimated by the ED.

(3) Action space of SUs: The action performed by each SU consists of the detection threshold, which is then used to decide whether the channel and sector pair is free or not. That is, the action taken by SU_i at time step t is $a_t^i = \epsilon_t^i(c_t^i, s_t^i)$ with $\epsilon_t^i(c_t^i, s_t^i) \geq 0$.

(4) Reward function: For each SU, the reward function can be designed according to two cases that could happen, which are: (#1) the sensing result and the actual state of the channel-sector pair are the same, (#2) the sensing result and the actual state of the channel-sector pair are different. The reward function should drive each agent to choose an action that provides an accurate sensing result. Therefore, the reward function is designed as follows:

$$r_t^i = \begin{cases} 0, & \text{if \#1 happens,} \\ -p & \text{if \#2 happens,} \end{cases} \tag{10}$$

where $p > 0$ is the penalty score.

3.2. MADDPG-assisted spectrum sensing scheme

To solve the Dec-POMDP, it is crucial to know the transition probability, which is a critical component of the model, but it is not known. In response to this challenge, we propose a solution called multi-agent distributed cooperative spectrum sensing (MA-DCSS).

Algorithm 1 MA-DCSS algorithm.

```
Initialize hyper-parameters: $\gamma, lra, lrc, \mathcal{B}, \vartheta, \sigma, \mu$
Initialize actor weight $\theta^{i,\mu}$, critic weight $\theta^{i,Q}$
Initialize target actor weight $\theta^{i,\mu_{tar}} \leftarrow \theta^{i,\mu}$, target critic weight
    $\theta^{i,Q_{tar}} \leftarrow \theta^{i,Q}$
repeat
    Obtain initial state $\mathbf{s}_0$, initial observation $\mathbf{o}_0$
    for $t := 1 \dots t_{max}$ do
        for $SU_i \in \mathcal{M}$ do
            Select action $a_t^i = \mu^i(o_t^i) + OU(\vartheta, \sigma, \mu)$
            $\epsilon_t^{i,k,l} = scale(a_t^i, min, max)$
        end for
        All SUs perform (3) based on actions and observations
        SUs obtain reward $\mathbf{r}_t$ based on (10)
        Environment transits to next state $\mathbf{s}_{t+1}$
        SUs observe next observation $\mathbf{o}_{t+1}$ by choosing a new channel
            and sector and calculating (2)
        Store $(\mathbf{o}_t, \mathbf{a}_t, \mathbf{o}_{t+1}, \mathbf{r}_t)$ in $\mathcal{B}$
        $\mathbf{s}_t \leftarrow \mathbf{s}_{t+1}, \mathbf{o}_t \leftarrow \mathbf{o}_{t+1}$
        for $SU_i \in \mathcal{M}$ do
            Sample random batch of $B$ samples from $\mathcal{B}$
            Update critic by (14) and (13)
            Update actor by (11)
            if update == true then
                Update target actor and target critic parameters
            end if
        end for
    end for
    Reset noise
until convergence or aborted
```

First, the proposed MA-DCSS adopts centralized training and decentralized execution (CTDE) architecture, in which the training phase (e.g. learning phase) and the execution phase are differently set up. More particularly, in the training phase, all agents are able to share their local knowledge with others so that they can learn from each other's experiences. In this phase, the critic network is trained in a centralized manner and fed with observations acknowledged by all agents. Meanwhile, during the execution phase, once all agents have gained sufficient knowledge about the environment, they can utilize the trained actor network to determine the optimal action based solely on their local observations. Second, the proposed MA-DCSS adopts the MADDPG algorithm proposed in [15]. MADDPG is a multi-agent extension of the DDPG algorithm [16], where each agent learns its own policy taking into account the policies of other agents in the environment. Specifically, each agent is equipped with one actor network which is used to learn its own individual policy. This network takes the current observation as the input and returns the action as the output. The actor network is updated using policy gradient, which is represented as:

$$\nabla_{\theta^{i,\mu}} J(\mu^i) = \mathbb{E}_{B\in\mathcal{B}}\Bigg[\nabla_\theta \mu^i(a_t^i|o_t^i)\nabla_{a_t^i} Q^i(\mathbf{o}_t, \mathbf{a}_t)|_{a_t^i=\mu^i(o_t^i)}\Bigg], \tag{11}$$

where B is the batch sample data, $\mathcal{B}$ denotes for the experience buffer, $\mathbf{a}_t = [a_t^i]_{i\in\mathcal{M}}$ represents the joint action of all agents, and $\mathbf{o}_t = [o_t^i]_{i\in\mathcal{M}}$ includes the partial observations collected from all agents. Moreover, $\mu^i(a_t^i|o_t^i)$ and $Q^i(\mathbf{o}_t, \mathbf{a}_t)$ respectively denote the deterministic policy and the centralized Q-value estimated function (e.g. the action-value function) of each agent. Besides the actor networks, all agents also have their critic networks which are learned together in the training phase. These critic networks take all the agents' observations $\mathbf{o}_t$ and the joint action $\mathbf{a}_t$ as input, and produce the Q-value as output. The loss function of the critic network is presented as:

$$\mathcal{L}(\theta^{i,Q}) = \mathbb{E}_{B\in\mathcal{B}}\left[\left(y - Q^i(\mathbf{o}_t, \mathbf{a}_t)\right)^2\right], \tag{12}$$

where y is given as:

$$y = r_t^i + \gamma Q^i_{tar}(\mathbf{o}_{t+1}, \mathbf{a}_{t+1})|_{a^i_{t+1}=\mu^i_{tar}\left(o^i_{t+1}\right)}, \tag{13}$$

where μ_{tar} and Q_{tar} are target actor and critic network, respectively.

During the training phase, the agent is faced with the important challenge of striking a balance between exploitation and exploration. To ensure sufficient exploration and enable the agent to discover the optimal point, noise is introduced into the actor's output. In this paper, we use the time-correlated Ornstein–Uhlenbeck (OU) noise [17] suggested by [16] for action exploration. The noise is defined as:

$$x_t = x_{t-1} + \vartheta(\mu - x_{t-1})dt + \sigma\mathcal{N}(0, dt), \tag{14}$$

where (ϑ, σ, μ) are noise parameters.

With these structures, the proposed MA-DCSS aims to train all of the agents to learn near-optimal detection thresholds that maximize the sensing performance. The proposed MA-DCSS algorithm for the training phase is illustrated in Algorithm 1.

3.3. Computational complexity analysis

During the training phase, each agent has its own actor and critic networks that perform forward and backpropagation to update the weights. Here, the main operations are matrix multiplications. Hence, the computational complexity for a single episode during the training phase can be expressed as $O\left(B\left(IN + (D-2)N^2 + JN\right)\right)$, where $D \geq 2$ is the number of layers, N is the hidden layer size, and I and J denote the dimensions of the input and output layers, respectively. In this work, $I = 4M$, where M is the number of SUs, and 4 is the total dimensions of the observation and action space. Moreover, J is the output Q-value and is equal to 1. On the other hand, in the execution phase, only forward propagation is needed for the actor-network, hence the computational complexity is $O\left((D-2)N^2\right)$. That is, it is easy to observe that the computational complexity is independent of the number of channels and sectors in both the training and execution phases. Instead, it grows linearly and in direct proportion to the number of SUs in the training phase exclusively. Therefore, it can be stated that the proposed algorithm possesses full scalability in relation to the number of channels, sectors, and SUs.

4. Simulation results

This section presents numerical experiments. One simulation scenario example is depicted in Fig. 2. We assumed 3 PUs and 5 SUs in the system. Each SU has three sectors for sensing and communication purposes, and the ability to select its directional beam in one of the sectors. All channels (e.g., c0, c1, c2) were locally and partially assigned to PUs, and each sector of SUs (e.g., s0, s1, s2) covers some PUs. For example, in Fig. 2, PU_0 is in use of channel 0 (c0), and SU_3 has PU_2 and PU_0 in its sector 0 (s0) and sector 2 (s2), respectively. We run the simulation 10 times with random SU and PU locations and average these results. The simulation parameters are summarized in Table 1.

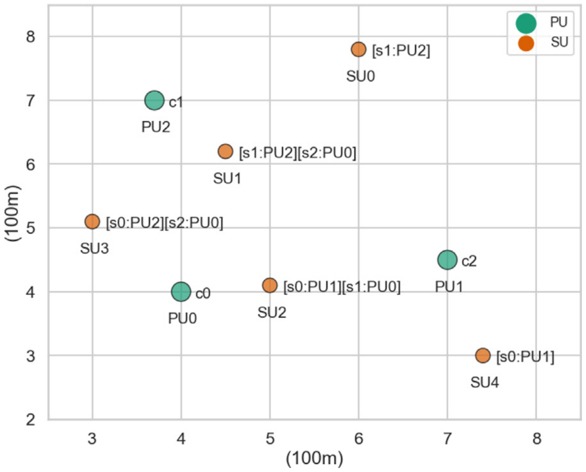

Fig. 2. One simulation scenario example.

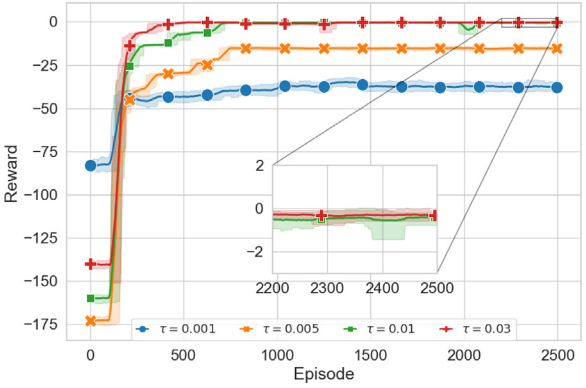

Fig. 3. Convergence with different sensing durations.

Convergence of the proposed algorithm: Fig. 3 shows the convergence of the proposed algorithm under different sensing durations τ ranging from 0.001s to 0.03s. First, it can be observed that the rewards attained by the agents increase as the duration of sensing time increases, but they stably converge, respectively. This positive correlation is attributed to the fact that a longer sensing duration enables the agents to gather more comprehensive information about the channel and sector being sensed. Second, it is worth noting that, beyond a certain threshold (e.g., $\tau = 0.01s$), the impact of further increasing the sensing time on overall performance becomes less significant.

Convergence comparison with baseline algorithms: Fig. 4 presents a convergence comparison between the proposed algorithm and the DDPG and PPO-based RL algorithms, which are both fully-distributed approaches and allow agents to be trained and executed independently. From this simulation, it can be observed that the proposed approach achieves faster

Table 1
Simulation parameters.

Parameters		Value
Number of PUs U		3
Number of channel bands K		3
Number of SUs M		5
Number of sectors L		3
SU's sensing range		300 m
Sensing duration τ		0.03 s
Operating frequency		60 GHz
Path loss model		$20\log_{10}\left(\frac{4\pi df}{c}\right)$
Transmit power		1 watt
α, β		0.99, 0.05
RL hyper-parameters		
Critic learning rate lrc		1e−3
Actor learning rate lra		1e−3
Discount factor γ		0.98
Number of layers D		3
Hidden layer size N		80
Number of time steps to update target networks		200
Penalty reward score p		5
MA-DCSS and DDPG	Number of time steps per episode t_{max}	64
	Noise parameters ϑ, σ, μ	0.15,1e−2,0
	Batch size B	8192
	Replay buffer size	5e3
PPO	Number of time steps per episode t_{max}	1000
	$sd_{init}, sd_{decay}, sd_{min}$	0.6,0.05,0.1
	Number of decay episodes	500
	ϵ_{clip}	0.2

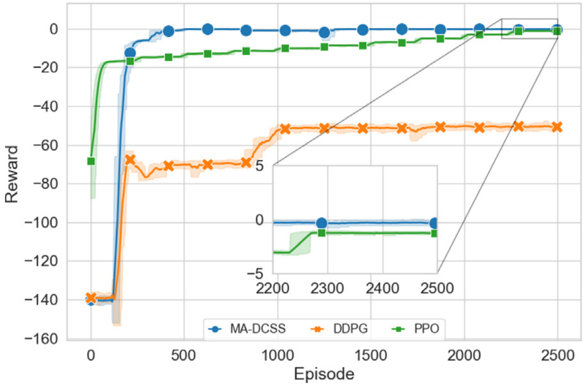

Fig. 4. Convergence comparison, $\tau = 0.03s$.

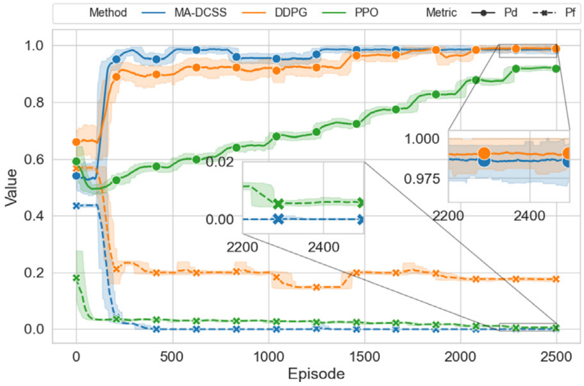

Fig. 5. P_d and P_f comparison, $\tau = 0.03s$.

convergence and attains a higher reward compared to the other comparing schemes. The PPO-based scheme provides similar (in fact, a little bit degraded) performance with the proposed approach, but it requires much more time to be converged. The DDPG-based scheme, on the other hand, provides the worst performance. This occurs due to the independent training of the agents, resulting in instability when there are frequent instances where at least one of the agents converges to a sub-optimal policy.

Detection probability and false alarm probability: Fig. 5 shows a performance comparison between the proposed algorithm with the DDPG and PPO-based RL algorithms in terms of P_d and P_f. The proposed algorithm provides approximately $P_d \approx 0.99$ detection probability and $P_f \approx 0$ false alarm probability after around 500 episodes. In contrast, the PPO-based RL algorithm provides approximately $P_d \approx 0.93$ of detection probability and $P_f \approx 0.007$ false alarm probability after around 2300 episodes. Regarding the DDPG-based RL algorithm, it performs exceptionally well in terms of detection probability, achieving $P_d \approx 1.0$. However, it yields the worst false alarm probability around $P_f \approx 0.2$.

5. Conclusion

This paper explores directional antenna-based cooperative spectrum sensing in fully distributed CRNs. First, we formulated a stochastic sequential optimization problem, which is an NP-hard, that optimizes the time-varying energy detection threshold to enhance detection probability and reduce false alarms. To tackle this in a distributed environment, we transform the problem into a Dec-POMDP problem. To find the optimal control of the Dec-POMDP in a realistic environment having no prior information on state–action transition probability, we developed the MADDPG-based reinforcement learning algorithm, called MA-DCSS. We analyzed that the proposed approach is scalable in terms of the number of channels, sectors, and SUs. It utilized the CTDE architecture to facilitate information sharing in the training phase and allow SUs to perform in a fully-distributed manner during the execution phase. Through simulations, it is shown that the proposed approach provides an enhanced convergence rate, accurate detection probabilities, and false alarm probabilities when compared to other learning methods. In future works, a more advanced distributed sensing algorithm that considers user mobility or the trade-off between sensing accuracy and transmission throughput.

CRediT authorship contribution statement

Thi Thu Hien Pham: Conceptualization, Methodology, Software, Validation, Writing – review & editing. **Wonjong Noh:** Supervision, Writing – review & editing. **Sungrae Cho:** Supervision, Reviewing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the Chung-Ang University, South Korea, Young Scientist Scholarship in 2022, in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (RS-2023-00209125), and in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2022-00156353) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

[1] I.F. Akyildiz, B.F. Lo, R. Balakrishnan, Cooperative spectrum sensing in cognitive radio networks: A survey, Phys. Commun. 4 (1) (2011) 40–62.

[2] R. Sarikhani, F. Keynia, Cooperative spectrum sensing meets machine learning: Deep reinforcement learning approach, IEEE Commun. Lett. 24 (7) (2020) 1459–1462.

[3] W. Na, J. Yoon, S. Cho, D. Griffith, N. Golmie, Centralized cooperative directional spectrum sensing for cognitive radio networks, IEEE Trans. Mob. Comput. 17 (6) (2018) 1260–1274.

[4] A. Gharib, W. Ejaz, M. Ibnkahla, Enhanced multiband multiuser cooperative spectrum sensing for distributed CRNs, IEEE Trans. Cogn. Commun. Netw. 6 (1) (2020) 256–270.

[5] A. Gao, C. Du, S. Ng, W. Liang, A cooperative spectrum sensing with multi-agent reinforcement learning approach in cognitive radio networks, IEEE Commun. Lett. PP (2021) 1.

[6] X. Tan, L. Zhou, H. Wang, Y. Sun, H. Zhao, B.-C. Seet, J. Wei, V.C. Leung, Cooperative multi-agent reinforcement-learning-based distributed dynamic spectrum access in cognitive radio networks, IEEE Internet Things J. 9 (19) (2022) 19477–19488.

[7] M.K. Giri, S. Majumder, Distributed dynamic spectrum access through multi-agent deep recurrent Q-learning in cognitive radio network, Phys. Commun. 58 (2023) 102054.

[8] Q. Wang, H.-N. Dai, O. Georgiou, Z. Shi, W. Zhang, Connectivity of underlay cognitive radio networks with directional antennas, IEEE Trans. Veh. Technol. 67 (8) (2018) 7003–7017.

[9] S. Tripathi, A.K. Gupta, S. Amuru, Coverage analysis of cognitive mmwave networks with directional sensing, in: 2021 55th Asilomar Conference on Signals, Systems, and Computers, IEEE, 2021, pp. 125–129.

[10] H. Urkowitz, Energy detection of unknown deterministic signals, Proc. IEEE 55 (4) (1967) 523–531.

[11] Y. Saleem, M.H. Rehmani, Primary radio user activity models for cognitive radio networks: A survey, J. Netw. Comput. Appl. 43 (2014) 1–16.

[12] N. Vlassis, M.L. Littman, D. Barber, On the computational complexity of stochastic controller optimization in POMDPs, ACM Trans. Comput. Theory (TOCT) 4 (4) (2012) 1–8.

[13] M. Mundhenk, J. Goldsmith, C. Lusena, E. Allender, Complexity of finite-horizon Markov decision process problems, J. ACM 47 (4) (2000) 681–720.

[14] K. Zhang, Z. Yang, T. Başar, Multi-agent reinforcement learning: A selective overview of theories and algorithms, Handb. Reinf. Learn. Control (2021) 321–384.

[15] R. Lowe, Y.I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, I. Mordatch, Multi-agent actor-critic for mixed cooperative-competitive environments, Adv. Neural Inf. Process. Syst. 30 (2017).

[16] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, 2015, arXiv preprint arXiv:1509.02971.

[17] G.E. Uhlenbeck, L.S. Ornstein, On the theory of the Brownian motion, Phys. Rev. 36 (5) (1930) 823.